

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Masaki ISHII, et al.

GAU:

SERIAL NO: NEW APPLICATION

EXAMINER:

FILED: HEREWITH

FOR: APPARATUS, METHOD, AND PROGRAM FOR DESIGNING REAL-TIME SYSTEM

REQUEST FOR PRIORITY

ASSISTANT COMMISSIONER FOR PATENTS
WASHINGTON, D.C. 20231

SIR:

- ☐ Full benefit of the filing date of U.S. Application Serial Number, filed, is claimed pursuant to the provisions of 35 U.S.C. §120.
- ☐ Full benefit of the filing date of U.S. Provisional Application Serial Number, filed, is claimed pursuant to the provisions of 35 U.S.C. §119(e).
- ☒ Applicants claim any right to priority from any earlier filed applications to which they may be entitled pursuant to the provisions of 35 U.S.C. §119, as noted below.

In the matter of the above-identified application for patent, notice is hereby given that the applicants claim as priority:

| <u>COUNTRY</u> | <u>APPLICATION NUMBER</u> | <u>MONTH/DAY/YEAR</u> |
|----------------|---------------------------|-----------------------|
| JAPAN | 2000-241688 | August 09, 2000 |

Certified copies of the corresponding Convention Application(s)

- ☒ are submitted herewith
- ☐ will be submitted prior to payment of the Final Fee
- ☐ were filed in prior application Serial No. filed
- ☐ were submitted to the International Bureau in PCT Application Number .
Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.
- ☐ (A) Application Serial No.(s) were filed in prior application Serial No. filed ; and
(B) Application Serial No.(s)
 - ☐ are submitted herewith
 - ☐ will be submitted prior to payment of the Final Fee

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.



Marvin J. Spivak

Registration No. 24,913

C. Irvin McClelland
Registration Number 21,124



22850

Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 10/98)

jc997 U.S. PTO
09/924489
08/09/01

Docket No. 212465US2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

INVENTOR(S) Masaki ISHII, et al.

SERIAL NO: New Application

FILING DATE: Herewith

FOR: APPARATUS, METHOD AND PROGRAM FOR DESIGNING REAL-TIME SYSTEM

jc997 U.S. PTO
09/924489
08/09/01

FEE TRANSMITTAL

ASSISTANT COMMISSIONER FOR PATENTS
WASHINGTON, D.C. 20231

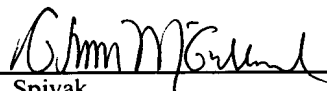
| FOR | NUMBER FILED | NUMBER EXTRA | RATE | CALCULATIONS |
|--|-----------------|-----------------|-----------|--------------|
| TOTAL CLAIMS | 18 - 20 = | 0 | × \$18 = | \$0.00 |
| INDEPENDENT CLAIMS | 6 - 3 = | 3 | × \$80 = | \$240.00 |
| <input type="checkbox"/> MULTIPLE DEPENDENT CLAIMS (If applicable) | | | + \$270 = | \$0.00 |
| <input checked="" type="checkbox"/> LATE FILING OF DECLARATION | | | + \$130 = | \$130.00 |
| BASIC FEE | | | | \$710.00 |
| TOTAL OF ABOVE CALCULATIONS | | | | \$1,080.00 |
| <input type="checkbox"/> REDUCTION BY 50% FOR FILING BY SMALL ENTITY | | | | \$0.00 |
| <input type="checkbox"/> FILING IN NON-ENGLISH LANGUAGE | | | + \$130 = | \$0.00 |
| <input type="checkbox"/> RECORDATION OF ASSIGNMENT | | | + \$40 = | \$0.00 |
| TOTAL | | | | \$1,080.00 |

- ☐ Please charge Deposit Account No. 15-0030 in the amount of A duplicate copy of this sheet is enclosed.
- ☒ A check in the amount of **\$1,080.00** to cover the filing fee is enclosed.
- ☒ The Commissioner is hereby authorized to charge any additional fees which may be required for the papers being filed herewith and for which no check is enclosed herewith, or credit any overpayment to Deposit Account No. 15-0030.
A duplicate copy of this sheet is enclosed.

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.

Date: 8/9/01


Marvin J. Spivak
Registration No. 24,913



22850

Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 10/00)

C. Irvin McClelland
Registration Number 21,124

日 本 国 特 許 庁
JAPAN PATENT OFFICE

S 699



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2000年 8月 9日

出 願 番 号

Application Number:

特願2000-241688

出 願 人

Applicant(s):

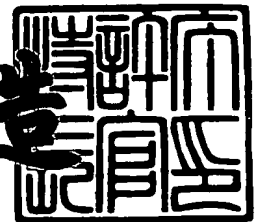
株式会社東芝

東芝エルエスアイシステムサポート株式会社

2001年 6月27日

特 許 庁 長 官
Commissioner,
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3060497

【書類名】 特許願

【整理番号】 46B003369

【提出日】 平成12年 8月 9日

【あて先】 特許庁長官殿

【国際特許分類】 G11C 11/00

【発明の名称】 リアルタイムシステム設計装置、リアルタイムシステム
設計方法及びリアルタイムシステム設計プログラムを記
録した記録媒体

【請求項の数】 9

【発明者】

 【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝
 マイクロエレクトロニクスセンター内

 【氏名】 石井 正樹

【発明者】

 【住所又は居所】 神奈川県川崎市幸区堀川町 5 8 0 番地 東芝エルエスア
 イシステムサポート株式会社内

 【氏名】 蒔田 寿樹

【特許出願人】

 【識別番号】 000003078

 【氏名又は名称】 株式会社 東芝

【特許出願人】

 【識別番号】 598010562

 【氏名又は名称】 東芝エルエスアイシステムサポート株式会社

【代理人】

 【識別番号】 100083806

 【弁理士】

 【氏名又は名称】 三好 秀和

 【電話番号】 03-3504-3075

【選任した代理人】

【識別番号】 100068342

【弁理士】

【氏名又は名称】 三好 保男

【選任した代理人】

【識別番号】 100100712

【弁理士】

【氏名又は名称】 岩▲崎▼ 幸邦

【選任した代理人】

【識別番号】 100100929

【弁理士】

【氏名又は名称】 川又 澄雄

【選任した代理人】

【識別番号】 100108707

【弁理士】

【氏名又は名称】 中村 友之

【選任した代理人】

【識別番号】 100095500

【弁理士】

【氏名又は名称】 伊藤 正和

【選任した代理人】

【識別番号】 100101247

【弁理士】

【氏名又は名称】 高橋 俊一

【選任した代理人】

【識別番号】 100098327

【弁理士】

【氏名又は名称】 高松 俊雄

【手数料の表示】

【予納台帳番号】 001982

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 リアルタイムシステム設計装置、リアルタイムシステム設計方法及びリアルタイムシステム設計プログラムを記録した記録媒体

【特許請求の範囲】

【請求項 1】 複数のタスクから構成されるリアルタイムシステムを設計する装置であって、

所定区間内に時系列的に並ぶ一連の前記タスクの動作が、非同期的に起こる要因によって影響を受けるか否かを判断する非同期要因影響判断部を具備することを特徴とするリアルタイムシステム設計装置。

【請求項 2】 前記非同期要因影響判断部が、非同期的に起こる要因によって影響を受けない区間であることを決定づけるパラメータを抽出するパラメータ抽出部を具備することを特徴とする請求項 1 記載のリアルタイムシステム設計装置。

【請求項 3】 複数のタスクから構成されるリアルタイムシステムを設計する装置であって、

非同期的に起こる要因によって影響を受けない一連の前記タスクを含む区間を複数個接続し、接続された一連の区間の動作に矛盾が起こらないか判断する区間接続・整合性判断部を具備することを特徴とするリアルタイムシステム設計装置。

。

【請求項 4】 複数のタスクから構成されるリアルタイムシステムを設計する方法であって、

所定区間内に時系列的に並ぶ一連の前記タスクの動作が、非同期的に起こる要因によって影響を受けるか否かを判断する非同期要因影響判断ステップを少なくとも含むことを特徴とするリアルタイムシステム設計方法。

【請求項 5】 前記非同期要因影響判断ステップが、非同期的に起こる要因によって影響を受けない区間であることを決定づけるパラメータを抽出するパラメータ抽出ステップを少なくとも含むことを特徴とする請求項 4 記載のリアルタイムシステム設計方法。

【請求項 6】 複数のタスクから構成されるリアルタイムシステムを設計す

る方法であって、

非同期的に起こる要因によって影響を受けない一連の前記タスクを含む区間を複数個接続し、接続された一連の区間の動作に矛盾が起こらないか判断する区間接続・整合性判断ステップを少なくとも含むことを特徴とするリアルタイムシステム設計方法。

【請求項 7】 複数のタスクから構成されるリアルタイムシステムを設計する方法をコンピュータに実行させるプログラムを記録した記録媒体であって、

所定区間内に時系列的に並ぶ一連の前記タスクの動作が、非同期的に起こる要因によって影響を受けるか否かを判断する非同期要因影響判断ステップ

を少なくとも含むことを特徴とするリアルタイムシステム設計プログラムを記録した記録媒体。

【請求項 8】 前記非同期要因影響判断ステップが、非同期的に起こる要因によって影響を受けない区間であることを決定づけるパラメータを抽出するパラメータ抽出ステップを少なくとも含むことを特徴とする請求項 7 記載のリアルタイムシステム設計プログラムを記録した記録媒体。

【請求項 9】 複数のタスクから構成されるリアルタイムシステムを設計する方法をコンピュータに実行させるプログラムを記録した記録媒体であって、

非同期的に起こる要因によって影響を受けない一連の前記タスクを含む区間を複数個接続し、接続された一連の区間の動作に矛盾が起こらないか判断する区間接続・整合性判断ステップを少なくとも含むことを特徴とするリアルタイムシステム設計プログラムを記録した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、リアルタイムシステムを設計する装置、リアルタイムシステムを設計する方法、及びリアルタイムシステムを設計する方法をコンピュータに実行させるプログラムを記録した記録媒体に関し、特にシステム全体の挙動の整合性を保ちながらを複数の並行動作を時系列的に記述してリアルタイムシステムを設計する技術に関する。

【0002】

【従来の技術】

複数のタスクが並行動作するリアルタイムシステムを構築する際、システム全体の動作の整合性を保った上で、各タスク間の同期通信制御を定義することは一般に困難である。なぜなら、例えば突発的に起こる割り込み処理や、一定時間間隔において処理が開始するタイムアウト処理などは、非同期的にタスクの動作に影響し、これを考慮したリアルタイムシステムの設計は、システムが大規模になるほど困難になるためである。

【0003】

リアルタイムシステムの設計を容易にするため、複数のタスクの各動作を時系列的に記述していく手法がすでに存在する。

【0004】

【発明が解決しようとする課題】

しかしながら、一般にリアルタイムシステムにおいては、各タスクの挙動は、前述した非同期的に起こる要因によりその挙動を変化させるため、単一の時系列情報のみでシステム全体の挙動を定義することは不可能である。

【0005】

このためリアルタイムシステム構築には、大変多くの時間を要し、その結果、リアルタイムシステムを使ったアプリケーションの開発期間が長期化していた。

【0006】

本発明は、このような課題を解決し、リアルタイムシステムを容易に設計することができるリアルタイムシステム設計装置、設計方法、および設計プログラムを記録した記録媒体を提供することを目的とする。

【0007】

【課題を解決するための手段】

上記課題を解決するため、本発明の特徴は、複数のタスクから構成されるリアルタイムシステムを設計する装置であって、所定区間内に時系列的に並ぶ一連の前記タスクの動作が、非同期的に起こる要因によって影響を受けるか否かを判断する非同期要因影響判断部を具備することにある。

【 0 0 0 8 】

「所定区間内に時系列的に並ぶ一連のタスクの動作」とは、例えば、1) タスクAがタスクBをD o r m a n t 状態からR e a d y 状態へ更新し、2) 次に、タスクAがタスクCをD o r m a n t 状態からR e a d y 状態へ更新し、3) 次に、タスクAが自タスクをR e a d y 状態からD o r m a n t 状態へ更新する、というような一連の流れをいう。

【 0 0 0 9 】

「非同期的に起こる要因」には、割り込み処理、タイムアウト処理が含まれる。割り込みが発生しても分岐が生じない場合は、非同期的に起こる要因によって影響を受けないと判断する。

【 0 0 1 0 】

本発明の特徴によって、ある区間内に時系列的に並ぶ一連のタスクの動作が非同期的に起こる要因によって影響を受けるか否かを判断することができ、影響を受けると判断された区間については、それに接続される区間として複数の区間を用意する必要があることがわかるようになる。

【 0 0 1 1 】

【発明の実施の形態】

以下、図面を参照して本発明の実施の形態について説明する。以下の図面の記載において、同一または類似の部分には同一または類似の符号が付してある。

【 0 0 1 2 】

本発明の実施形態として、ユーザがリアルタイムOSを使用してリアルタイムシステムを構築する例を説明する。このために、本発明ではシステムの振る舞いを“シナリオ”という概念で時系列に表す。また、ある限られた区間のシナリオを“小シナリオ”として複数用意する。そして、“小シナリオ”を不整合がないように結合していくことで、システムの様々な振る舞いを定義する。以下にその詳細を述べる。なお、リアルタイムOSは μ ITRON 3.0仕様、またはこれと同様仕様のリアルタイムOS仕様に準拠したものを言い、以下、リアルタイムシステムの説明に用いられる技術用語の定義は、すべて μ ITRON 3.0仕様に従うものとする。

【0013】

図1に、一つのシナリオでリアルタイムシステムの挙動を定義する例を示す。図1に示すように、本実施形態ではリアルタイムシステムの振る舞いを時系列的に定義していく方法を採用。定義された時系列情報をここではシナリオと呼び、以下のように定義する。

【0014】

「シナリオ」とは、システムを構成する各タスクの、持ちうる状態の変化及びそれぞれの相互作用を時系列的に定義したものである。

【0015】

前記の如く、一般に、システム動作の条件が変化する場合は考慮すると、割り込み処理などの影響のため、一つのシナリオのみでシステム全体の挙動を定義することは不可能である。例えば、図1のT1の時点において、ある割り込みが起動したときに、タスクBではなく、タスクAに処理を移すような挙動は単一のシナリオでは定義できない。

【0016】

そこで、動作中のシステムのある動作局面に着目して、その場合に起こりうる各タスクの振る舞いを、システム開発者（ユーザ）が短いシナリオとして定義する。同様に様々な局面について各タスクの振る舞いを短いシナリオとして定義する。そして、定義した複数の短いシナリオを図2に示すように結合し、システム全体の振る舞いを定義する。この短いシナリオを「小シナリオ」と呼ぶ。

【0017】

つまり、「小シナリオ」とは、シナリオから、各タスク動作が外的要因の影響を受けない区間を切り出したものである。例えば、割り込み等の非同期事象が起こっても、その区間に分岐が起こらない区間などである。

【0018】

小シナリオの区間は、上記定義の条件を満足する限り延長可能である。しかし、一つの小シナリオを適当な位置で二つに分割し、後半の小シナリオをループ処理の起点にする場合など、システムの定義内容に応じて、適宜小シナリオを分割することも可能である。

【 0 0 1 9 】

図 3 は、小シナリオのパラメータ及びそれを利用しておこなう小シナリオ接続時の整合性判断を説明するための図である。

【 0 0 2 0 】

図 3 に示すように、小シナリオは、(1) 開始時点での各タスクの状態 (Ready, Wait 等)、(2) 完了時点での各タスクの状態、(3) 小シナリオ内でアクセスされるリアルタイム OS 資源 (フラグ、セマフォ等) 及び大域変数などにより決定づけることができる。本実施形態で説明する装置は、各タスクの状態等を「小シナリオのパラメータ」として、すべての小シナリオについて抽出する機構を有する。

【 0 0 2 1 】

また、本実施形態の装置は、全ての小シナリオを結合してシステムを完成させるときに、ある小シナリオの完了状態のパラメータと、それに結合させる小シナリオの開始状態のパラメータを照らし合わせて、矛盾なくシステムが動作するかどうかを判断する機構を有する。

【 0 0 2 2 】

これらの機構により、もし非同期的に起こる要因により、リアルタイム OS 資源及び大域変数の内容が変化しても、小シナリオ単位ではその影響を受けず、小シナリオの組合せをもってそれらの変化に対応した分岐処理を定義できるため、整合性を保つことが容易になる。

【 0 0 2 3 】

図 4 は、本発明の実施形態にかかるリアルタイムシステム設計装置を説明する機能ブロック図である。図 4 に示すように、本実施形態のリアルタイムシステム設計装置は、ユーザインタフェース部 11 と、小シナリオパラメータ抽出部 12 と、小シナリオ接続部・整合性判断部 13 と、小シナリオ状態情報記憶部 14 と、小シナリオ間接続データ記憶部 15 と、を少なくとも含む。

【 0 0 2 4 】

ユーザインタフェース部 11 は、ユーザからの入力データを受け付ける。この入力データには、タスクやリアルタイム OS 資源・大域変数に関する情報、及び

小シナリオを定義するための各タスクの処理（システムコール発行、大域変数の読み書き等）が含まれる。また、ユーザが定義した小シナリオのパラメータを表示し、小シナリオ内のタスクの動作・状態遷移の様子を図3に示した様に、時系列的に表示する機能も有する。

【0025】

小シナリオパラメータ抽出部12は、ユーザがユーザインタフェース部11に入力した小シナリオ開始時のパラメータや小シナリオ内の各タスクの処理情報を受け取り、（1）小シナリオが入力された処理をもって終了するか否かを判断する、（2）定義した小シナリオからパラメータを抽出する、という2つの機能を持つ。また、判断、抽出結果は、ユーザインタフェース部11へ出力され、ユーザへ提示される。抽出されたパラメータは、小シナリオ状態情報記憶部14に書き込まれる。

【0026】

小シナリオ接続部・整合性判断部13は、ユーザインタフェース部11に入力された複数の小シナリオのうち、ユーザが選択した2個の小シナリオの接続情報を受け取り、（1）前段の小シナリオと後段の小シナリオを接続することでシステムが矛盾した動作を起こさないかという「システムの動作整合性」を判断する、（2）小シナリオの完了部を、前段の非同期的に起こる要因を含む小シナリオの開始部に接続することでループ状態が作られた場合は、関係する全ての小シナリオのパラメータを小シナリオの状態情報記憶部14より読み出し、それらの情報を総合してシステムの動作整合性を判断する、という2つの機能を持つ。割り込みやタイムアウト処理が定義された場合もこれと同様である。動作整合性の判断結果はユーザインタフェース部11へ出力する。小シナリオの接続情報は、小シナリオ間接続データ記憶部15に書き込む。

【0027】

小シナリオ状態情報記憶部14は、ユーザが定義した小シナリオ群から、小シナリオのパラメータ抽出部12により抽出されたそれぞれの開始・完了状態パラメータを保存しておく機能を有する。

【0028】

小シナリオ間接続データ記憶部 15 は、小シナリオ接続部・整合性判断部 13 により出力される小シナリオ間の接続データ、及び非同期的に起こる要因がどの小シナリオへ影響しうるかという情報を保存しておく機能を有する。

【0029】

以下に、リアルタイムOSを使用したアプリケーションを作成し、そのシステム全体を小シナリオで記述するまでの一連の動作を説明する。想定するリアルタイムOSは μ ITRON 3.0仕様に準拠したものである。

【0030】

アプリケーションの概要は、以下のとおりである。

- 1) 電話機のダイヤル発信動作を行う。
- 2) 使用者の数字キー入力を受け取り、これに対応したダイヤルのパルス信号を回線に発信する。

【0031】

アプリケーション仕様は、以下のとおりである。

- 1) 電話機には、数字以外のキーも備えられているが、ダイヤル発信時にそのキーが押された場合は無効であるとして何も処理をしない。
- 2) ダイヤル発信はパルス列を用いてソフトウェアにより行う。
- 3) ダイヤル発信中も、次の数字キー入力を受け付ける。

システムの構成は、以下のとおりである。

使用タスク： タスク S (優先度 1) … スタートアップ処理用

タスク A (優先度 2) … ダイヤル発信用

タスク B (優先度 3) … キー入力受付用

使用OS資源： メッセージバッファ (1 個)

使用データ： キー入力結果 (数字)

本実施形態で使用される μ ITRON 3.0仕様システム：

sta__tsk, ext__tsk, snd__mbf, rcv__mbf, dly__tsk

本実施形態で定義されるシステムの制限： 1 優先度に 1 タスクのみ設定

[小シナリオ定義の流れ]

＜全小シナリオに共通するパラメータの設定＞

ユーザはまず、図4中のユーザインタフェース部11に、システムに含まれるタスク、各タスクの優先度、資源を入力する。さらにシステム起動時にReady状態となるタスクも入力する。

【0032】

本実施形態では、1)上記のタスクS、A及びB、2)タスクS、A及びBの優先度、3)資源としてメッセージバッファ、4)システム起動時にReady状態となるタスクとしてタスクS、を入力する。ユーザインタフェース部11は小シナリオのパラメータ抽出部12へ入力された情報を送る。

【0033】

図5に小シナリオパラメータ抽出部12の動作フローを、図6に開始状態決定処理(b)の動作フローを、図7にタスク動作登録処理(c)部の動作フローを、図8にパラメータ(データ)抽出処理(d)の動作フローを、図9にパラメータ(タスク状態)抽出処理(e)の動作フローを、図10に現シナリオ整合性判断処理(f)の動作フローを、図11に小シナリオ接続の動作フローを、図12～図16に小シナリオ1～5の定義を、図17に小シナリオ接続の概念図それぞれ示す。

【0034】

以下に、図12に示す小シナリオ1の定義について説明する。小シナリオ1は、「タスクSがタスクA及びBを起動する処理」に関する小シナリオである。

【0035】

初期設定(a)では、ユーザインタフェース部11が、ユーザの入力内容を受け取る(図5、ステップS20)。

【0036】

開始状態決定処理(b)では、小シナリオパラメータ抽出部12は、ユーザインタフェース部11から受け取った情報の中から、開始状態を表すパラメータを抽出する(図5、ステップS30)。具体的には、タスクSはReady、他のタスクはDormantであり、アクセス資源としてメッセージバッファを持つ、という内容を抽出する。そして、その内容をユーザインタフェース部11へ出

力する。また、抽出された開始状態パラメータは、小シナリオ状態情報記憶部14に記憶される。

【0037】

タスク動作登録処理(c)では、タスクSの次の動作をユーザインタフェース部11に問い合わせる(図5、ステップS40)。ユーザインタフェース部11は、各タスク状態を全てユーザに提示し、ユーザによるタスクSの動作登録を受け付ける。小シナリオ1の定義として、ユーザは、タスクAに対するシステムコールsta__tskと、タスクBに対するシステムコールsta__tskとを発行し、その後ext__tskを発行する処理を登録する。その手順は次の通りである。

【0038】

<タスクSのsta__tsk(タスクA)発行を登録するまで>

タスクSがReady状態である。つまり、Ready状態のタスクが1つ以上存在する。よって、図7のステップS41はYESへ分岐する。

Ready状態のタスクのうち最も優先度の高いタスクとして、タスクSが検索される(ステップS42)。

ユーザインタフェース部11に対しタスク動作定義情報の問い合わせがなされる(ステップS43)。ユーザは、ユーザインタフェース部11を通してタスクSの動作として“sta__tsk(タスクA)”を入力する。

図8に示すパラメータ(データ)抽出処理(d)へ進む。

【0039】

登録したタスク(sta__tsk)は、資源・データにアクセスしないので、ステップS51はNOへ分岐する。

図9に示すパラメータ(タスク状態)抽出処理(e)へ進む。

【0040】

登録したタスク(sta__tsk)は、wai__sem等に該当しないので、ステップS610はNOへ分岐する。

登録したタスクは、sta__tskであるから、ステップS620はYESへ分岐する。

自タスク（タスク S）より高い優先度のタスクは存在しないので、ステップ S 621 は NO へ分岐し、ユーザインタフェース部へ小シナリオ定義終了か問い合わせる（ステップ S 640）。

小シナリオの定義を終了しないので、ステップ S 641 は NO へ分岐する。

図 7 に示すタスク動作登録処理（c）へ戻る。

【0041】

＜タスク S の `sta__task`（タスク A）発行を登録してから、`sta__task`（タスク B）発行を登録するまで＞

タスク S 及びタスク A が Ready 状態である。つまり、Ready 状態のタスクが 1 つ以上存在する。よって、図 7 のステップ S 41 は YES へ分岐する。

Ready 状態のタスクのうち最も優先度の高いタスクとして、タスク S が検索される（ステップ S 42）。

ユーザインタフェース部 11 に対しタスク動作定義情報の問い合わせがなされる（ステップ S 43）。ユーザは、ユーザインタフェース部 11 を通してタスク S の動作として“`sta__task`（タスク B）”を入力する。

図 8 に示すパラメータ（データ）抽出処理（d）へ進む。

【0042】

登録したタスク（`sta__task`）は、資源・データにアクセスしないので、ステップ S 51 は NO へ分岐する。

図 9 に示すパラメータ（タスク状態）抽出処理（e）へ進む。

【0043】

登録したタスク（`sta__task`）は、`wai__sem`等に該当しないので、ステップ S 610 は NO へ分岐する。

登録したタスクは、`sta__task`であるから、ステップ S 620 は YES へ分岐する。

自タスク（タスク S）より高い優先度のタスクは存在しないので、ステップ S 621 は NO へ分岐し、ユーザインタフェース部へ小シナリオ定義終了か問い合わせる（ステップ S 640）。

小シナリオの定義を終了しないので、ステップ S 641 は NO へ分岐する。

図7に示すタスク動作登録処理(c)へ戻る。

【0044】

＜タスクSのsta__tsk(タスクB)発行を登録してから、ext__tsk発行を登録するまで＞

タスクS、タスクA及びタスクBがReady状態である。つまり、Ready状態のタスクが1つ以上存在する。よって、ステップS41はYESへ分岐する。

Ready状態のタスクのうち最も優先度の高いタスクとして、タスクSが検索される(ステップS42)。

ユーザインタフェース部11に対しタスク動作定義情報の問い合わせがなされる(ステップS43)。ユーザは、ユーザインタフェース部11を通してタスクSの動作として“ext__tsk”を入力する。

図8に示すパラメータ(データ)抽出処理(d)へ進む。

【0045】

登録したタスク(ext__tsk)は、資源・データにアクセスしないので、ステップS51はNOへ分岐する。

図9に示すパラメータ(タスク状態)抽出処理(e)へ進む。

【0046】

登録したタスクがext__tskであるから、ステップS610はYESへ分岐する。

登録したタスクがext__tskであるから、自タスク(タスクS)をDormantとする(ステップS611)。

自タスク(タスクS)より低い優先度のタスクの中にReady状態のものが存在する(タスクA、BはともにReady状態である)ため、ステップS612はYESへ分岐し、ユーザインタフェース部へ小シナリオ定義終了か問い合わせる(ステップS640)。

小シナリオの定義を終了しないので、ステップS641はNOへ分岐する。

図7に示すタスク動作登録処理(c)へ戻る。

【0047】

＜タスクSのext__tsk発行を登録してから、タスクAのrcv__mbf発行を登録するまで＞

タスクA及びタスクBがReady状態である。つまり、Ready状態のタスクが1つ以上存在する。よって、ステップS41はYESへ分岐する。

Ready状態のタスクのうち最も優先度の高いタスクとして、タスクAが検索される（ステップS42）。

ユーザインタフェース部11に対しタスク動作定義情報の問い合わせがなされる（ステップS43）。ユーザは、ユーザインタフェース部11を通してタスクAの動作として“rcv__mbf”を入力する。

図8に示すパラメータ（データ）抽出処理（d）へ進む。

【0048】

登録したタスク（rcv__mbf）は、資源・データにアクセスするので、ステップS51はYESへ分岐する。

アクセスする資源としてmbf（メッセージバッファ）を抽出する（ステップS52）。

現シナリオはdly__tskを発行したことはないので、ステップS53はNOへ分岐する。

アクセスの結果シナリオを分岐させないため、ステップS55はNOへ分岐する。

小シナリオの定義を終了しないため、ステップS58はNOへ分岐する。

図9に示すパラメータ（タスク状態）抽出処理（e）へ進む。

【0049】

登録したタスクがrcv__mbfであるから、ステップS610はYESへ分岐する。

登録したタスクがrcv__mbfであるから、自タスク（タスクA）を受信対象（mbf）に対するWait状態とする（ステップS611）。

自タスク（タスクA）より低い優先度のタスクBがReady状態なので、ステップS612はYESへ分岐し、ユーザインタフェース部へ小シナリオ定義終了か問い合わせる（ステップS640）。

小シナリオの定義を終了するので、ステップS641はYESへ分岐する。

図10に示す現シナリオ整合性判断処理(f)へ進む。

【0050】

完了状態パラメータは抽出されていないので、ステップS71はNOへ分岐する。

ユーザインタフェース部11へタスクの待ち対象資源を追加するか問い合わせる(ステップS73)。

Wait状態タスクの待ち対象資源を追加しないため、ステップS75はNOへ分岐する。

抽出されたパラメータ(タスクSがDormant、タスクAがWait(mbf)、タスクBがReady、アクセスする資源はメッセージバッファ)を現シナリオの完了パラメータとして確定する(ステップS77)。

以上で、図12に示す小シナリオ1の定義が完了する。

【0051】

なお、小シナリオの一部を変更したり、新たに挿入したりする場合、変更又は挿入する小シナリオに接続される前段及び後段の小シナリオは既に決定されている。つまり、変更又は挿入する小シナリオの開始状態及び完了状態は既に決定されている。図10のステップS71がYESに分岐するのは、既に後段シナリオが決定されている場合のことを指す。

【0052】

<ユーザが、タスクBにキー入力結果の有効・無効を判断させる処理を定義する場合>

次に、図13に示す小シナリオ2の定義について説明する。小シナリオ2は、タスクBが、1)電話使用者からのキー入力を受け付け、2)入力結果が有効か無効かを判定する処理に関する。小シナリオ2は、条件判断による分岐が存在するため、完了状態において2通りの状態を持つ。この点が、小シナリオ2と小シナリオ1との大きな相違点である。

【0053】

まず、パラメータ抽出部12は、図6に示すフロー図に沿って開始状態決定処

理を行う。

現シナリオ（小シナリオ2）はシステムの開始部ではないため、ステップS32はNOへ分岐する。

現シナリオ（小シナリオ2）はd1y起床タスクのものではないため、ステップS33もNOへ分岐する。

前段の小シナリオ（小シナリオ1）の完了状態を、現シナリオ（小シナリオ2）の開始状態としてユーザインタフェース部11へ出力する（ステップS34）。

前段シナリオ（小シナリオ1）の完了状態は分岐ではないため、ステップS5はNOへ分岐する。

資源待ち状態のタスク（タスクA）に対して、受信可能な資源が開始状態では存在しないため、ステップS36はNOへ分岐する。

現段階では後段シナリオ（小シナリオ3）が定義されていないため、ステップS37はNOへ分岐する。

図7に示すタスク動作登録（c）へ進む。

【0054】

Ready状態のタスクBが存在するため、ステップS41はYESへ分岐する。

Ready状態のタスクのうち最も優先度の高いタスクとしてタスクBを検索する（ステップS42）。

ユーザインタフェース部11へタスクBの動作定義情報を問い合わせる（ステップS43）。

図8に示すパラメータ（データ）抽出処理（d）へ進む。

【0055】

登録したタスク動作（「キー入力」）は資源・データにアクセスしないためステップS51はNOへ分岐する。

図9に示すパラメータ（タスク状態）抽出処理（e）へ進む。

【0056】

登録したタスク動作はwai__sem等ではないため、ステップS610はN

〇へ分岐する。

登録したタスク動作は `sig__sem` 等ではないため、ステップ S620 は N
〇へ分岐する。

登録したタスク動作は `dly__tsk` ではないため、ステップ S630 は N
〇へ分岐する。

ユーザインタフェース部 11 へ小シナリオ定義終了か問い合わせる（ステップ
S640）。

小シナリオの定義を終了しないため、ステップ S641 は N〇へ分岐する。

図 7 に示すタスク動作登録処理（c）へ進む。

【0057】

Ready 状態のタスク B が存在するため、ステップ S41 は YES へ分岐す
る。

Ready 状態のタスクのうち最も優先度の高いタスクとしてタスク B を検索
する（ステップ S42）。

ユーザインタフェース部 11 へタスク B の動作定義情報を問い合わせる（ステ
ップ S43）。

図 8 に示すパラメータ（データ）抽出処理（d）へ進む。

【0058】

登録したタスク動作（「入力判定」）は資源・データにアクセスするためステ
ップ S51 は YES へ分岐し、アクセスする資源・データを抽出する（ステップ
S52）。

現シナリオ（小シナリオ 2）は `dly` 起床タスクではないためステップ S53
は N〇へ分岐する。

アクセスの結果、シナリオを分岐させるためステップ S55 は YES へ分岐し
、ユーザインタフェース部へ何通りの分岐か問い合わせる（ステップ S56）、
分岐数及び分岐先を決めるパラメータを抽出する（ステップ S57）。本実施形
態では、ユーザは分岐の数がキー入力の「有効」または「無効」の 2 通り、その
判断要因はデータ「キー入力結果」であることを入力する。

図 10 に示す現シナリオ整合性判断処理（f）へ進む。

【0059】

完了状態パラメータは抽出されていないので、ステップS71はNOへ分岐する。

ユーザインタフェース部11へタスクの待ち対象資源を追加するか問い合わせる（ステップS73）。

Wait状態タスクの待ち対象資源を追加する場合は、ステップS75はYESへ分岐する。この場合は、追加しようとする資源を現シナリオ（小シナリオ2）内でアクセスしないことを確認し、待ち対象を追加する（ステップS76）。

【0060】

タスクAは、何ら処理を定義していないので、タスク状態Wait (mbf) に待ち対象を追加しても問題はない。よって、ユーザは、dly__tskによるWait状態をタスクAに対する開始・完了状態パラメータとして追加する。

以上で、図13に示す小シナリオ2の定義を完了する。

【0061】

＜ユーザが、タスクBにOS資源をアクセスする処理を定義する場合＞

次に、図14に示す小シナリオ3の定義について説明する。小シナリオ3は、タスクBがキー入力結果の内容を送信するためにOS資源をアクセスする処理に関する。

【0062】

前段の小シナリオ2でキー入力判定が「有効」である場合は、メッセージバッファからの受信を待っているタスクAにキー入力を送信するため、タスクBがsnd__mhfを発行する。

【0063】

一方、小シナリオ2でキー入力判定が「無効」であった場合は、タスクBは有効なキー入力があるまでループ処理し続ける。このため、小シナリオ2は同じ小シナリオ2の開始状態へ結合する。結合については後述する。

【0064】

小シナリオ3の定義時には、パラメータ抽出部12の開始状態決定処理（b）が「前段シナリオの完了状態は分岐する」と判断し（図6、ステップS35でY

ES)、分岐時の判断要因を「キー入力結果」、その内容を「有効」として抽出し(ステップS38)、他のタスク状態と共に開始状態パラメータとして反映する。

【0065】

以下、小シナリオ1定義時と同様に、ユーザはタスクBがsnd__mbfを発行する処理を入力する。このとき、図8に示すパラメータ(データ状態)抽出処理(d)では、「登録したタスク動作は資源・データにアクセスするか？」(ステップS51)はYESに分岐する。

現シナリオ(小シナリオ3)はdly__tskを発行したことはないので、「現シナリオはdly起床タスクか？」(ステップS53)はNOに分岐する。

現シナリオ(小シナリオ3)ではメッセージバッファへのアクセスによってシナリオが分岐することはないので、「アクセスの結果、シナリオを分岐させるか？」(ステップS55)はNOへ分岐する。

「小シナリオの定義を終了するか？」(ステップS58)では、ユーザはメッセージバッファから受信後のタスクAの動作をループ構造の開始点とするために、YESを選択する。なお、ここで小シナリオを終了しないで続けてタスクAの処理を定義した場合でも、後から必要に応じて小シナリオの分割を行うことは可能である。

【0066】

<ユーザが、タスクAに非同期的処理(ここでは、タイムアウト処理)を定義する場合>

次に、図15に示す小シナリオ4の定義について説明する。小シナリオ4の内容は、タスクAがダイヤル出力を行い、出力中も電話使用者からのキー入力を受け付けるためにタイムアウト処理をおこなうというものである。

【0067】

まず、図6に示す開始状態決定処理(b)を行う。

現シナリオ(小シナリオ4)はシステムの開始部ではない(ステップS32、NO)。また、現シナリオ(小シナリオ4)はdly起床タスクのものでもない(ステップS33、NO)。

前段シナリオ（小シナリオ3）の完了状態を現シナリオ（小シナリオ4）の開始状態として抽出する（ステップS34）。つまり、「タスクSがDormant、タスクAがWait(mbf)、タスクBがReady」が小シナリオ4の開始状態として抽出される。

前段シナリオ（小シナリオ3）の完了状態は分岐である（ステップS35、YES）。

分岐時の条件判断に用いられたデータ・資源とその内容を抽出する（ステップS38）。つまり、「mbf=有、キー入力結果=有効」が抽出される。

資源待ち状態のタスク（タスクA）に対して、受信可能な資源（mbf）が開始状態で既に存在する（ステップS36、YES）。

資源待ちタスク（タスクA）をReady状態に、資源（mbf）の状態を送信後の状態に更新する（ステップS39）。

まだ、後段シナリオ（小シナリオ5）が定義されていないので（ステップS37、NO）、図7に示すタスク動作登録処理（c）へ移る。

【0068】

以下、小シナリオ1定義時と同様に、ユーザはタスクAに対し、受信したメッセージに対応したダイヤルを出力する処理を入力し、続いて出力が終わるまでの一定時間Wait状態とするためdly_tskを発行する処理を入力する。

タスク動作登録後、図8に示すパラメータ（データ）抽出処理（d）へ移る。

登録したタスク動作（dly_tsk）は資源・データにアクセスしないので（ステップS51、NO）、図9に示すパラメータ（タスク状態）抽出処理へ移る。

【0069】

図9において「登録したタスク動作はdly_tskシステムコールか？」（ステップS630）は、YESに分岐する。タスクAの状態をWait(dly_tsk)に更新し（ステップS631）、現シナリオ整合性判断処理（f）へ移行する。

【0070】

<ユーザが、非同期的処理においてアクセスする資源・データを定義する場合>

次に、図16に示す小シナリオ5について説明する。小シナリオ5は、タスクAがd l y__t s kによる待ち状態からタイムアウトした後、OS資源をアクセスする小シナリオ5を定義する流れを説明する。

【0071】

現シナリオはd l y起床タスクであるから（図6、ステップS33、YES）、d l y起床タスク（すなわちタスクA）をR e a d y状態、他のタスク状態をd o n' t c a r eとして抽出し、開始状態とする。

図7に示すタスク動作登録（c）へ移る。

【0072】

以下、小シナリオ1定義時と同様に、ユーザは、タスクAの動作を登録する。登録の内容は「タスクAが、メッセージバッファを介したダイヤル情報の受信を待つため、r c v__m b fを発行する」というものである。タスクAは非同期的に起床するタスクであるため、r c v__m b f発行後、システムのどの状態でメッセージバッファが変化してもそれに対応できるような処理を定義する必要がある。

【0073】

その整合性を判断するために、図8に示すパラメータ（データ）抽出処理（d）の「アクセスする資源・データを非同期変化データとして抽出」の処理（ステップS54）で、メッセージバッファを非同期変化データとして小シナリオ状態情報記憶部14に書き込む。

【0074】

その後、図9に示すパラメータ（タスク状態）抽出処理（e）の「自タスクより低い優先度のタスクの中にR e a d y状態のものがあるか？」（ステップS612）でNOへ分岐して、小シナリオ5の定義を完了し、図10に示す現シナリオ整合性判断処理（f）へ移行する。

【0075】

以上により、小シナリオ1～5の定義が完了する。

【0076】

[小シナリオ間接続、リアルタイムシステム構築の流れ]

図11に小シナリオ接続処理のフロー図を、図17に小シナリオ接続の概念図を示す。図11に示す処理フローに沿って小シナリオを接続することによってリアルタイムシステムが矛盾なく動作する。小シナリオ接続部・整合性判断部13(図4)が、これまで抽出されたパラメータから、「リアルタイムシステムが矛盾なく動作する」と判断する流れを以下に説明する。なお、図17中の(1)～(5)はそれぞれ小シナリオ(1)～小シナリオ(5)に対応する。また、小シナリオ5は非同期的に起こる小シナリオであるため、接続される矢印を点線で表している。

【0077】

小シナリオ1に小シナリオ2を接続する際、シナリオ接続部・整合性判断部13は、1)後段シナリオ(シナリオ2)の開始部と前段シナリオ(シナリオ1)の終了部の各タスク状態(データ・資源があればそれらも)を全て読み込み、2)両者の間に対応するパラメータを比較して、3)後段シナリオの開始状態パラメータと、前段シナリオの終了パラメータが一致するかを判断する(ステップS803)。この場合は全て一致するのでYESと判断する。

続いて、後段シナリオとなるシナリオ2の既定義内容から、「接続によりループ構造が形成されるか?」(ステップS804)はNOへ分岐し、「後段シナリオにdly_tsk発行があるか?」(ステップS805)もNOへ分岐し、「後段シナリオは非同期変化データをアクセスするか?」(ステップS806)もNOへ分岐する、というフローを通り、後段シナリオの接続確定・接続データを出力する(ステップS807)。

以上の流れは、図11において、S801→S802→S803→S804→S805→S806→S807→S808のように処理される。

【0078】

小シナリオ2は完了状態を2つ持つため、小シナリオ2に接続する小シナリオとしては各完了状態に対応する2個の小シナリオをユーザが選択しなければならない(ステップS802)。

【0079】

小シナリオ2の完了状態「キー入力結果」が「無効」の場合、タスクBをルー

プさせ、キー入力を待ち受ける処理を定義する。このため、ユーザは小シナリオ2の完了状態と開始状態を接続する。開始・完了状態を比較すると、完了状態の「キー入力結果」が開始状態では存在しない。この場合、装置は開始状態に存在するパラメータを元に、それらに対応する前段シナリオの完了状態パラメータとの一致を見て（この場合は各タスクの状態の一致のみを判断）接続を確定する。

以上の流れは、図11において、S801→S802→S803→S804→S815→S807→S808のように処理される。

【0080】

小シナリオ2の完了状態「キー入力結果」が「有効」の場合、小シナリオ3を次に接続し、キー入力結果をメッセージバッファに出力する処理を行う。この接続は前述の小シナリオ1, 2接続時と同様の流れで実現される。つまり、図11において、S801→S802→S803→S804→S805→S806→S807→S808のように処理される。

【0081】

小シナリオ3と4の接続も同様であるが、小シナリオ4ではタスクAがシステムコールdly__tskを発行する。つまり、「後段シナリオにdly__tsk発行タスクがあるか？」（ステップS805）はYESへ分岐し、次段でdly__tsk発行位置を抽出する（ステップS814）。

小シナリオパラメータ抽出部12（図4）で非同期変化データとしてメッセージバッファが抽出されており、小シナリオ4はメッセージバッファからの受信処理を含むため、「後段シナリオは非同期変化データをアクセスするか？」（ステップS806）はYesへ分岐し、判断結果をユーザインタフェース部へ出力する（ステップS812）。この段階ではメッセージバッファの内容の非同期的な変化は起こらないと判断し、「接続する」の判断を入力する（ステップS813、YES）。

以上の流れは、図11において、S801→S802→S803→S804→S805→S814→S806→S812→S813のように処理される。

【0082】

次に、小シナリオ4を小シナリオ2へ接続する場合について説明する。小シナ

リオ4の完了部は、ダイヤル発信後次のキー入力を待ち受けるため、ユーザは小シナリオ2の開始部へ接続する。この接続も前述の小シナリオ1と2の接続時と同様の流れで実現される。図11に示す小シナリオ接続フローにおいて「接続によりループ構造が形成されるか?」(ステップS804)はYESとなる。また、小シナリオ3が非同期変化データ(ここではメッセージバッファ)をアクセスしているため、「ループ構造の中で、非同期変化データをアクセスする小シナリオはあるか?」(ステップS815)もYESとなる。そして、小シナリオ3においてタスクAがタスクBに影響を及ぼす可能性があることをユーザインタフェース部を通してユーザに示す(ステップS817)。その後小シナリオ4と小シナリオ2の接続が確定する(ステップS807)。

以上の流れは、図11において、S801→S802→S803→S804→S815→S816→S817→S807→S808→S818のように処理される。

【0083】

小シナリオ5は、dly__tskによる待ち状態からタイムアウトして起床した場面から開始する。そのため、前段にどのシナリオが接続されるかは決定できず、後続く小シナリオのみを接続することになる。「ユーザインタフェース部へ後段シナリオ選択問い合わせ」(ステップS802)では、dly__tsk発行後実行されうる小シナリオを、読み込んだ全シナリオのパラメータ中から選出し、ユーザに提示する。この例では、dly__tsk発行後実行されうる小シナリオは小シナリオ2, 3, 4が該当し、それらの内、その開始状態が小シナリオ5の完了状態と一致する小シナリオ2, 3, 4が後段に接続されると装置が判断し、ユーザインタフェース部11を通してユーザに示す。

【0084】

以上のようにして、図17に示すような小シナリオの接続が完了し、リアルタイムシステムとして完成する。

【0085】

各小シナリオは、非同期的に起きる割り込み等によって分岐することはなく、また、小シナリオの完了状態とそれに接続する小シナリオの開始状態をチェック

し、矛盾が生じないことを確認して各接続を実施しているので、システム全体としてい矛盾は生じない。

【0086】

また、上述のリアルタイムシステム設計方法において必要とされる各ステップをコンピュータに実行させるプログラムを記録媒体に記録し、これをコンピュータ読み取らせ、実行させることにより、上述したリアルタイムシステム設計方法を実現することができる。ここで、記録媒体としては、例えばメモリ装置、磁気ディスク装置、光ディスク装置、磁気テープ装置などのプログラムを記録できるような装置が含まれる。

【0087】

図18は、これら記録媒体に格納されたプログラムを読み取り、そこに記述されたステップに従ってリピータ挿入を実現するコンピュータシステムの一例を示す概観図である。このコンピュータシステム80の本体前面には、フロッピーディスクドライブ81及びCD-ROMドライブ82が設けられており、磁気ディスク装置としてのフロッピーディスク83又は光ディスク装置としてのCD-ROM84を各ドライブ入口から挿入し、所定の読み出し操作を行うことにより、これらの記録媒体に格納されたプログラムをシステム内にインストールすることができる。また、所定のドライブ装置を接続することにより、例えばゲームパックなどに使用されているメモリ装置としてのROM85や磁気テープ装置としてのカセットテープ86を用いることもできる。

【0088】

【発明の効果】

上記の如く、本発明を使用することにより、並行動作単位が複数個動作するリアルタイムシステムの設計時に、全体のシステムの動作の整合性を保った上で、それぞれの並行動作単位間の同期通信制御を容易に定義することができる。そして、リアルタイムシステムを利用したアプリケーションの開発期間が短縮できる。

【図面の簡単な説明】

【図1】

一つのシナリオでリアルタイムシステムの挙動を定義する例を示す図である。

【図 2】

複数のシナリオを組み合わせてリアルタイムシステムの挙動を定義する例を示す図である。

【図 3】

小シナリオのパラメータ及びそれを利用して行う小シナリオ接続時の整合性判断を説明するための図である。

【図 4】

本発明の実施形態を概念的に説明する機能ブロック図である。

【図 5】

図 4 に示す小シナリオパラメータ抽出部 1 2 における動作の概略を示すフローチャートである。

【図 6】

図 5 に示す開始状態決定処理 (b) の動作を示すフローチャートである。

【図 7】

図 5 に示すタスク動作登録処理 (c) の動作を示すフローチャートである。

【図 8】

図 5 に示すパラメータ (データ) 抽出処理 (d) の動作を示すフローチャートである。

【図 9】

図 5 に示すパラメータ (タスク状態) 抽出処理 (e) の動作を示すフローチャートである。

【図 10】

図 5 に示す現シナリオ整合性判断処理 (f) の動作を示すフローチャートである。

【図 11】

図 4 に示す小シナリオ接続部・整合性判定部 1 3 における動作を示すフローチャートである。

【図 12】

「タスク S がタスク A 及び B を起動する処理」を定義する小シナリオ 1 を示す図である。

【図 13】

「タスク B にキー入力結果の有効・無効を判断させる処理」を定義する小シナリオ 2 を示す図である。

【図 14】

「タスク B に OS 資源をアクセスさせる処理」を定義する小シナリオ 3 を示す図である。

【図 15】

「タスク A に非同期的な動作をさせる処理」を定義する小シナリオ 4 を示す図である。

【図 16】

「タスク A が非同期的な動作においてアクセスする資源・データ」を定義する小シナリオ 5 を示す図である。

【図 17】

小シナリオを接続して構築したリアルタイムシステムを示す図である。

【図 18】

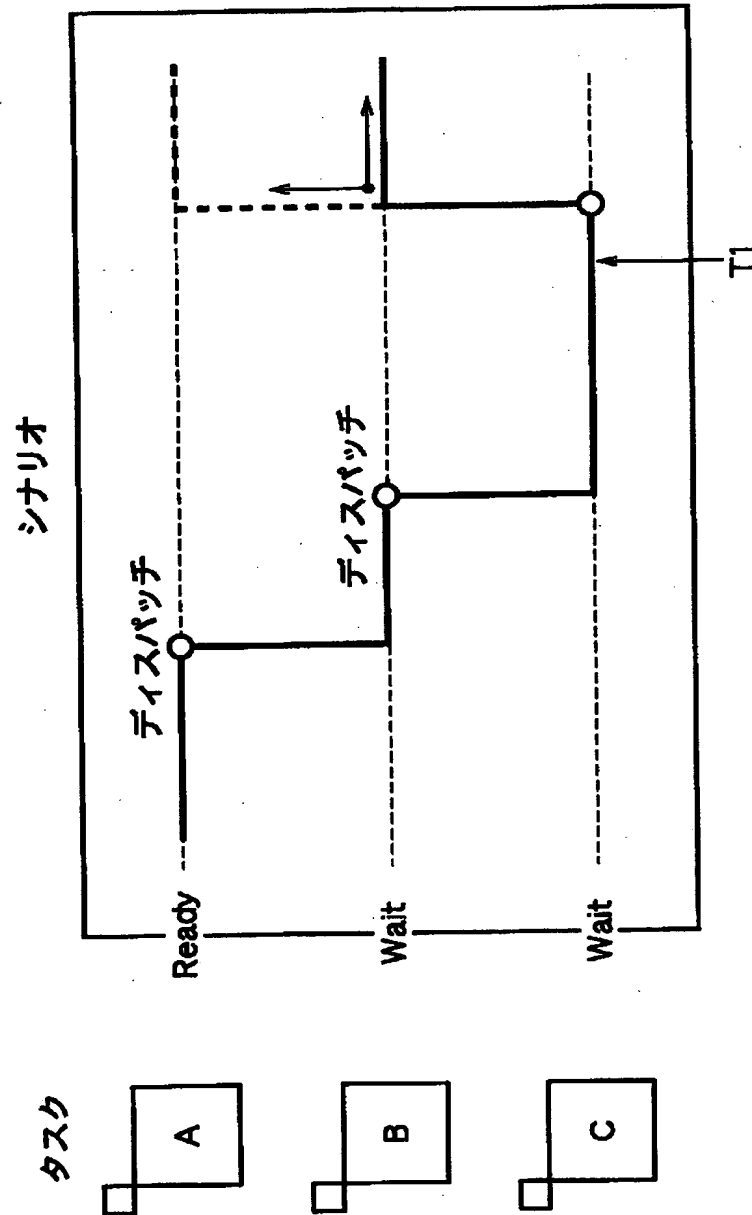
本発明のリアルタイムシステム設計方法を実現するコンピュータシステムの一例を示す概観図である。

【符号の説明】

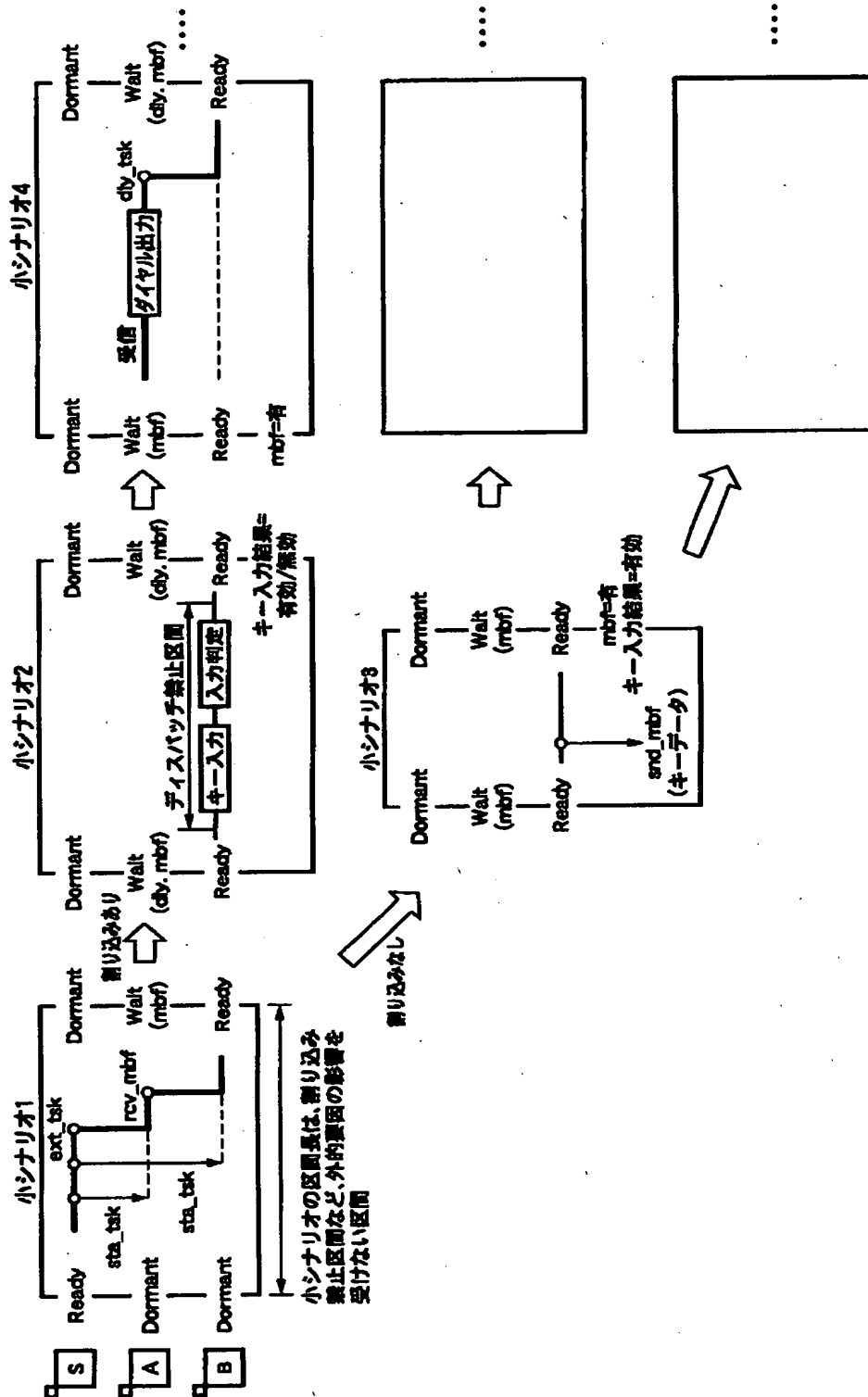
- 11 ユーザインタフェース部
- 12 小シナリオパラメータ抽出部
- 13 小シナリオ接続部・整合性判断部
- 14 小シナリオ状態情報記憶部
- 15 小シナリオ間接続データ記憶部

【書類名】 図面

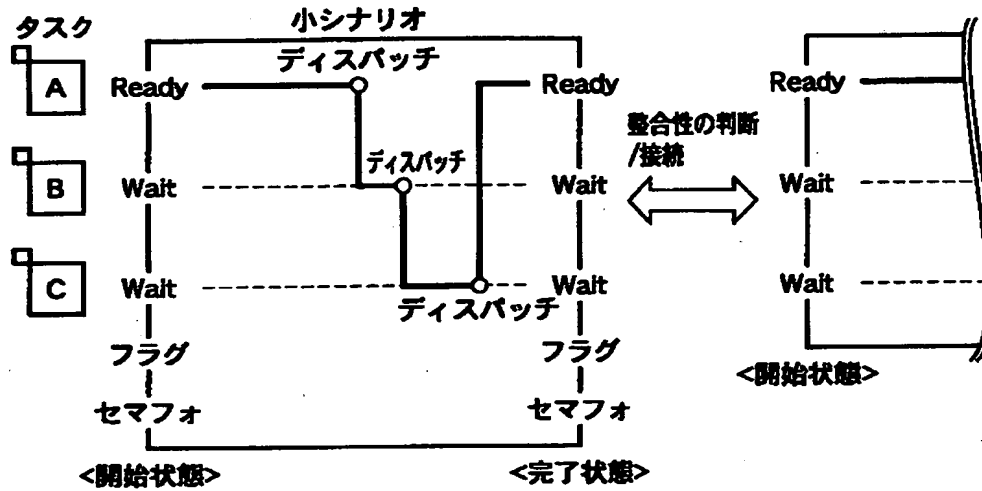
【図 1】



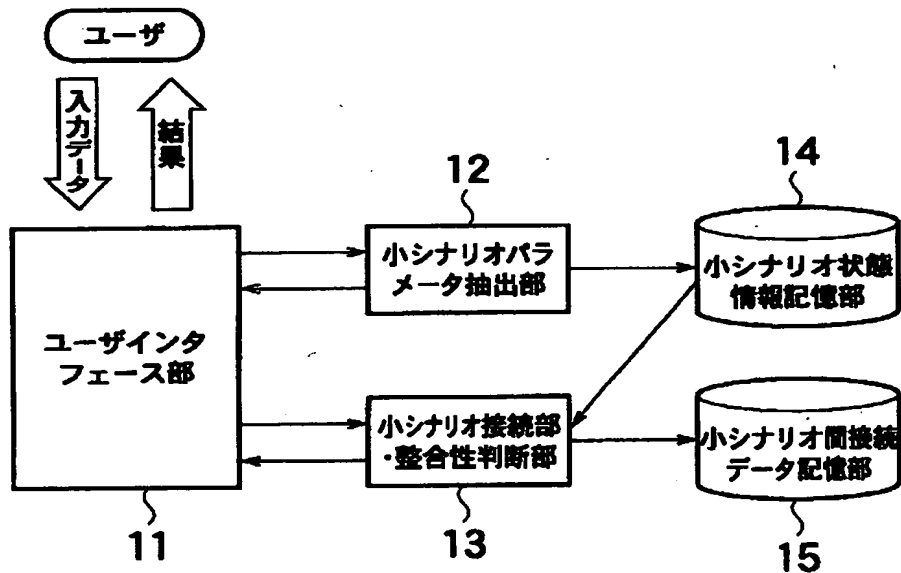
【図 2】



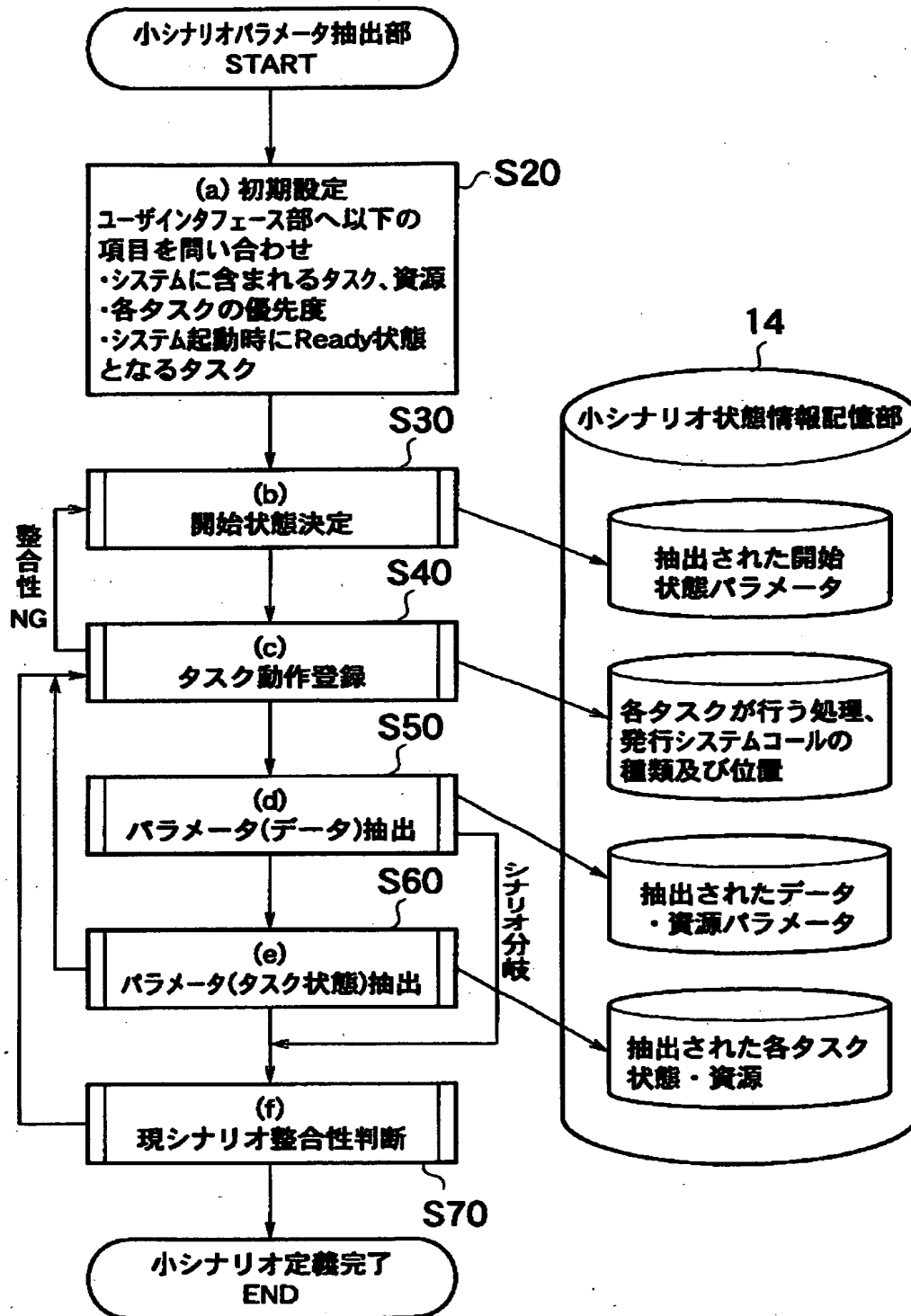
【図 3】



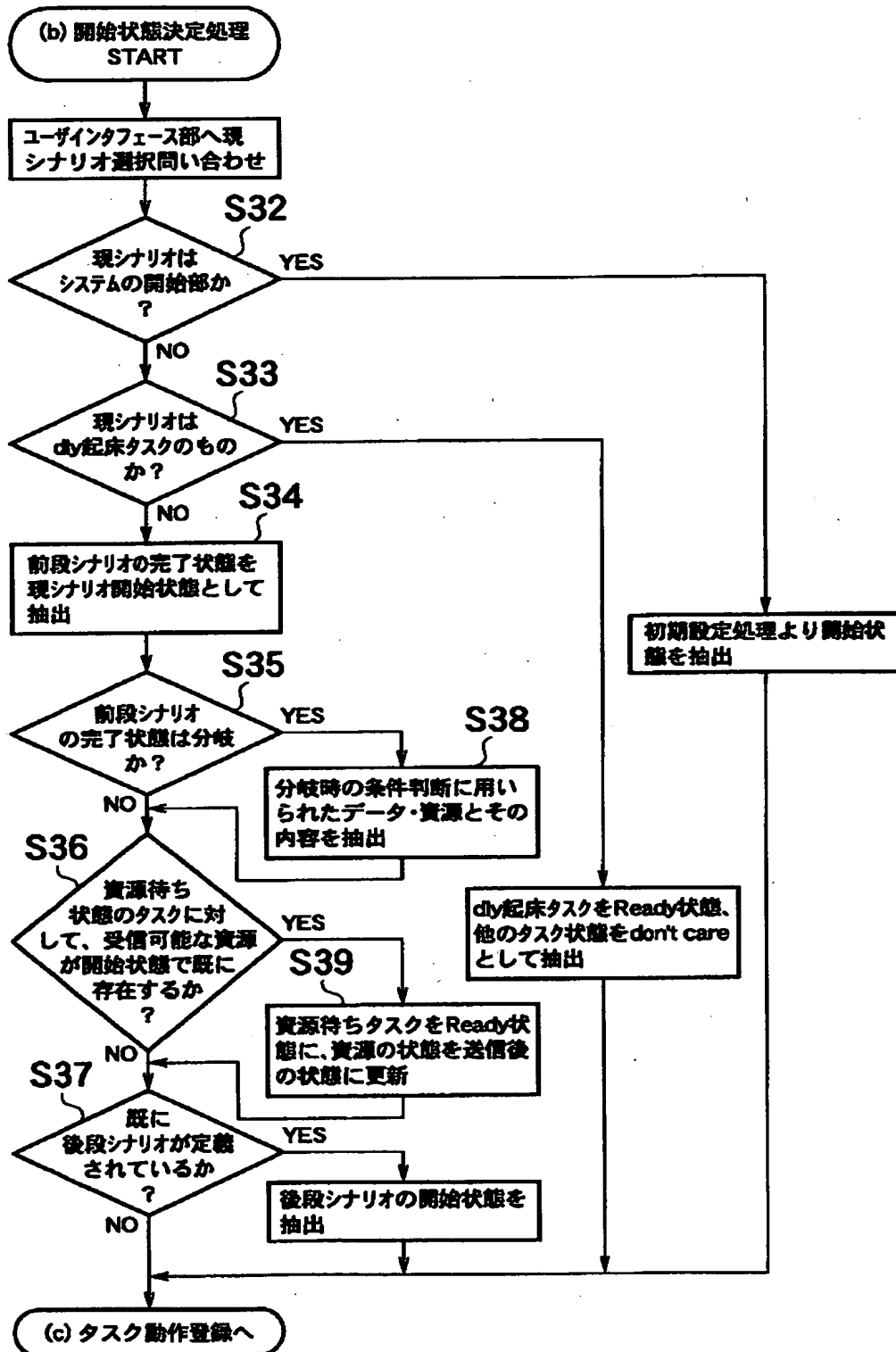
【図 4】



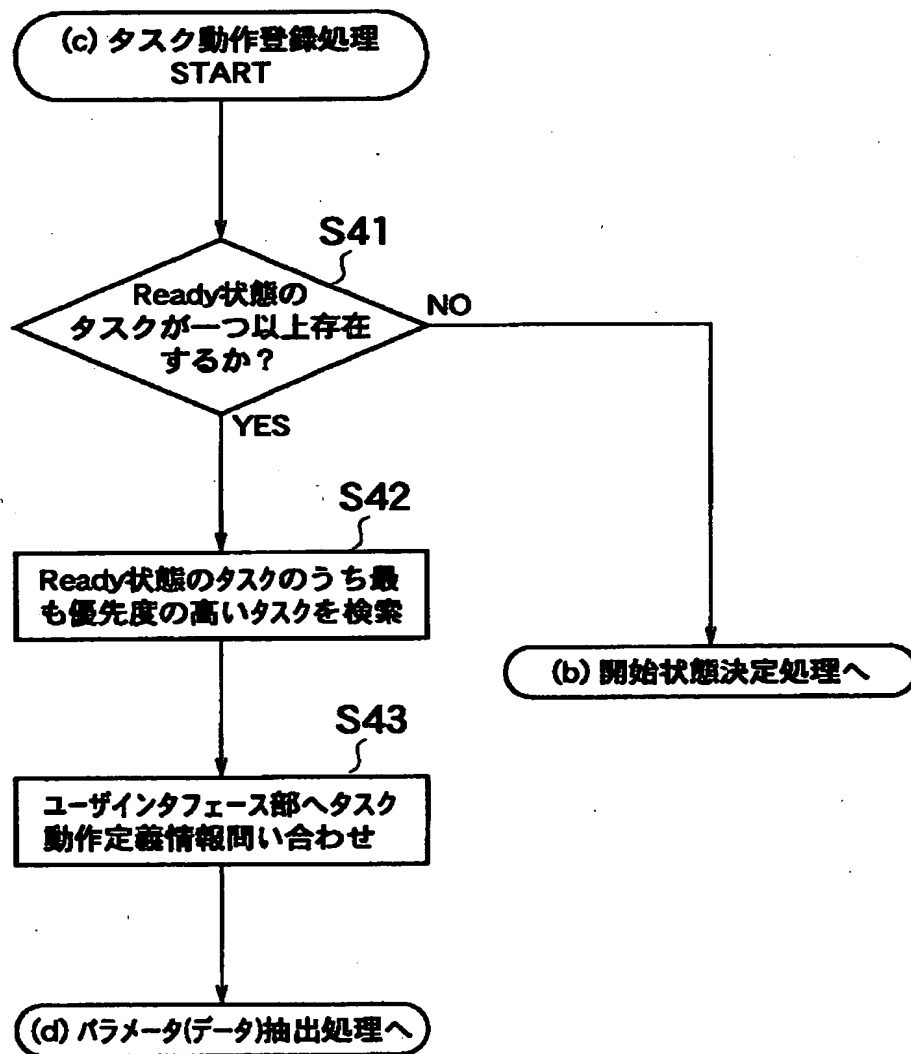
【図 5】



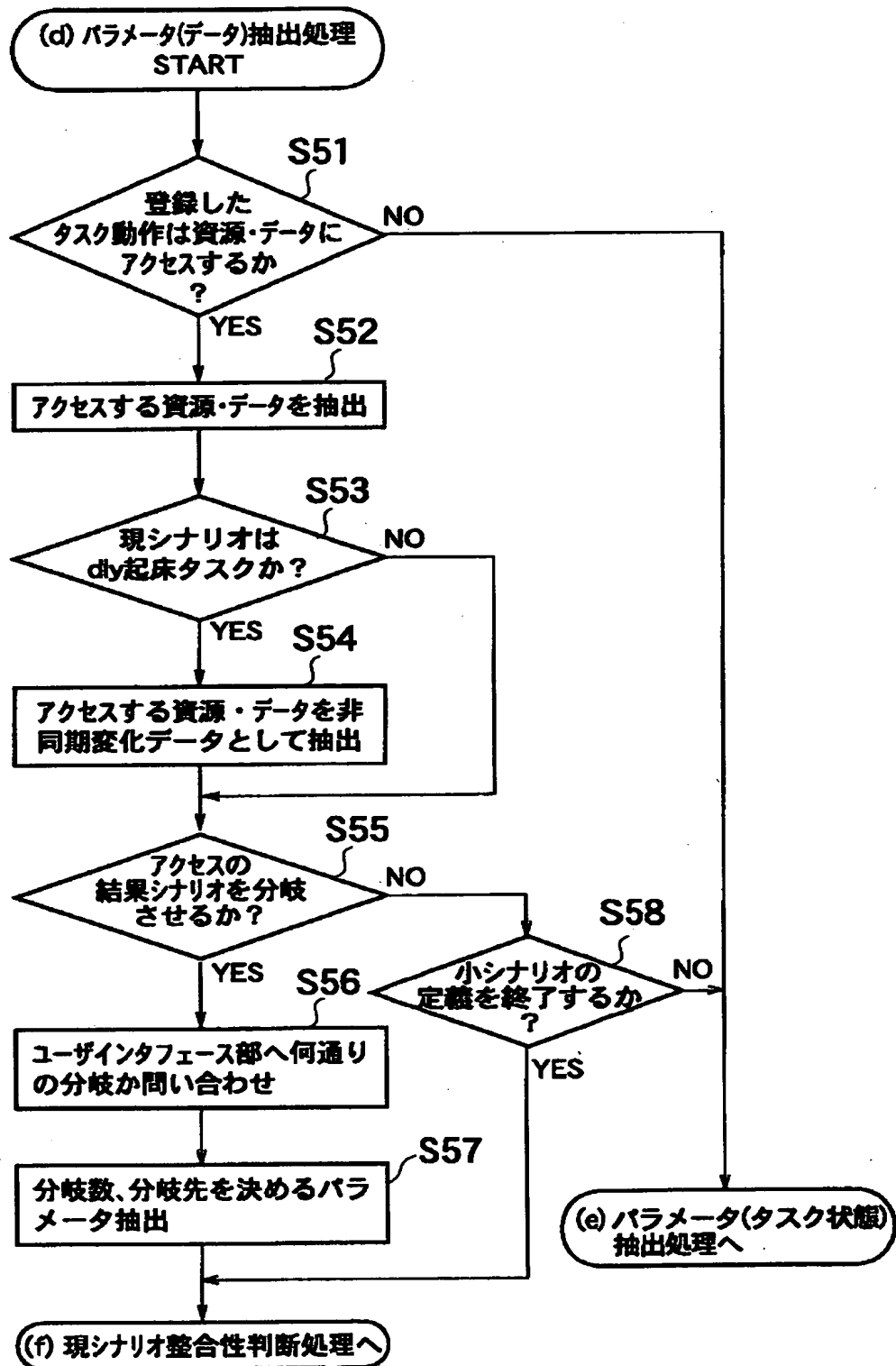
【図 6】



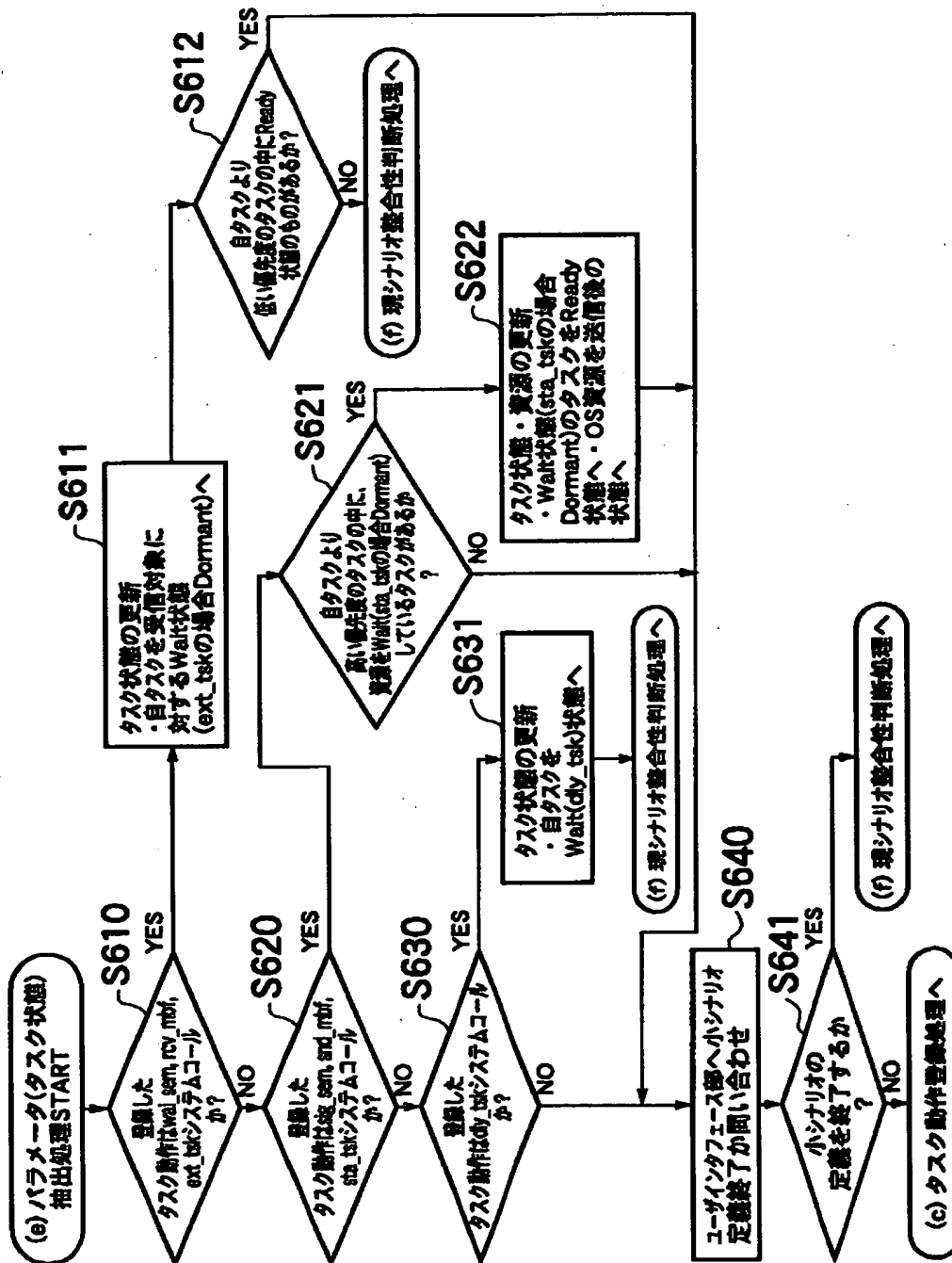
【図 7】



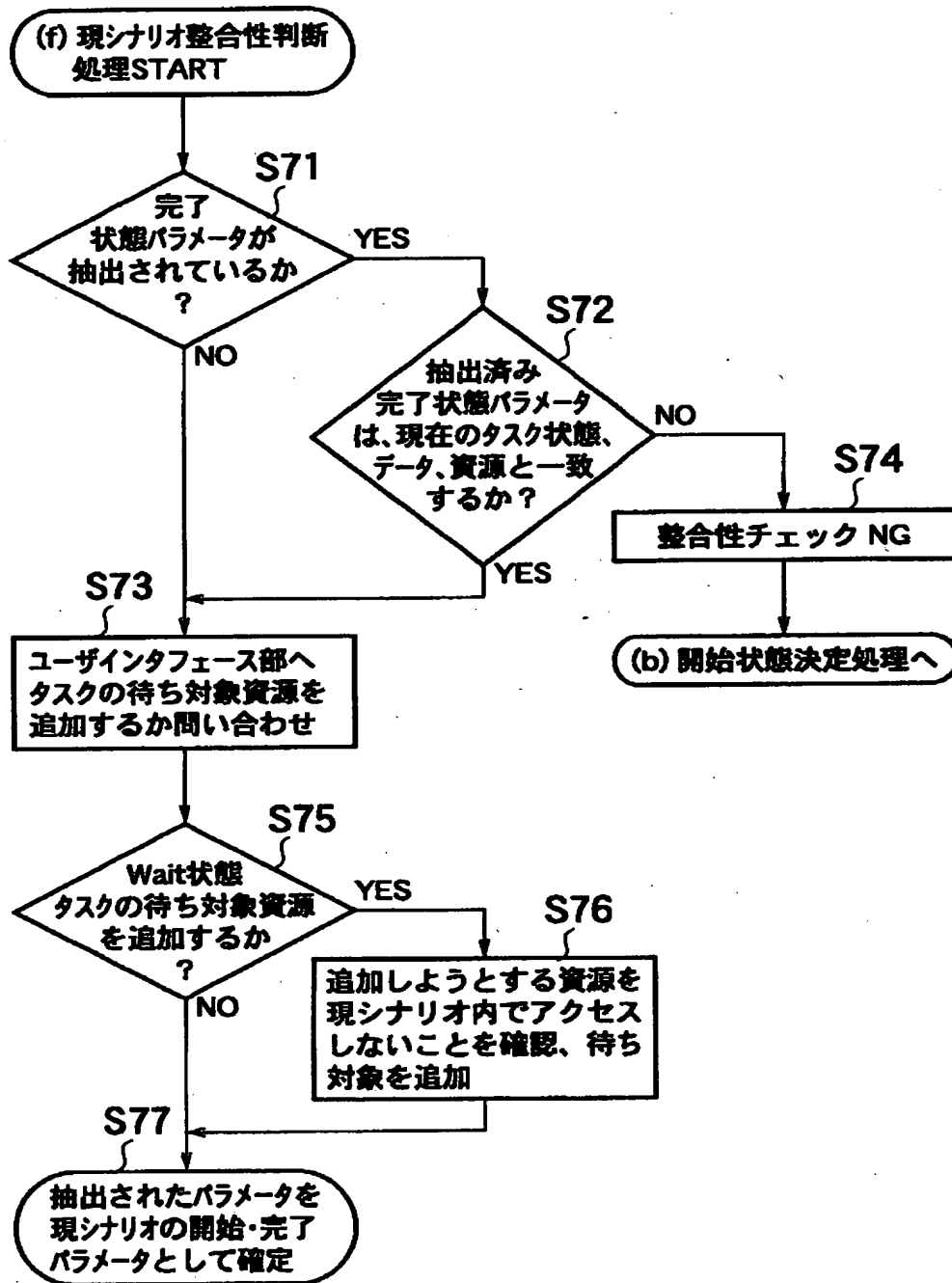
【図 8】



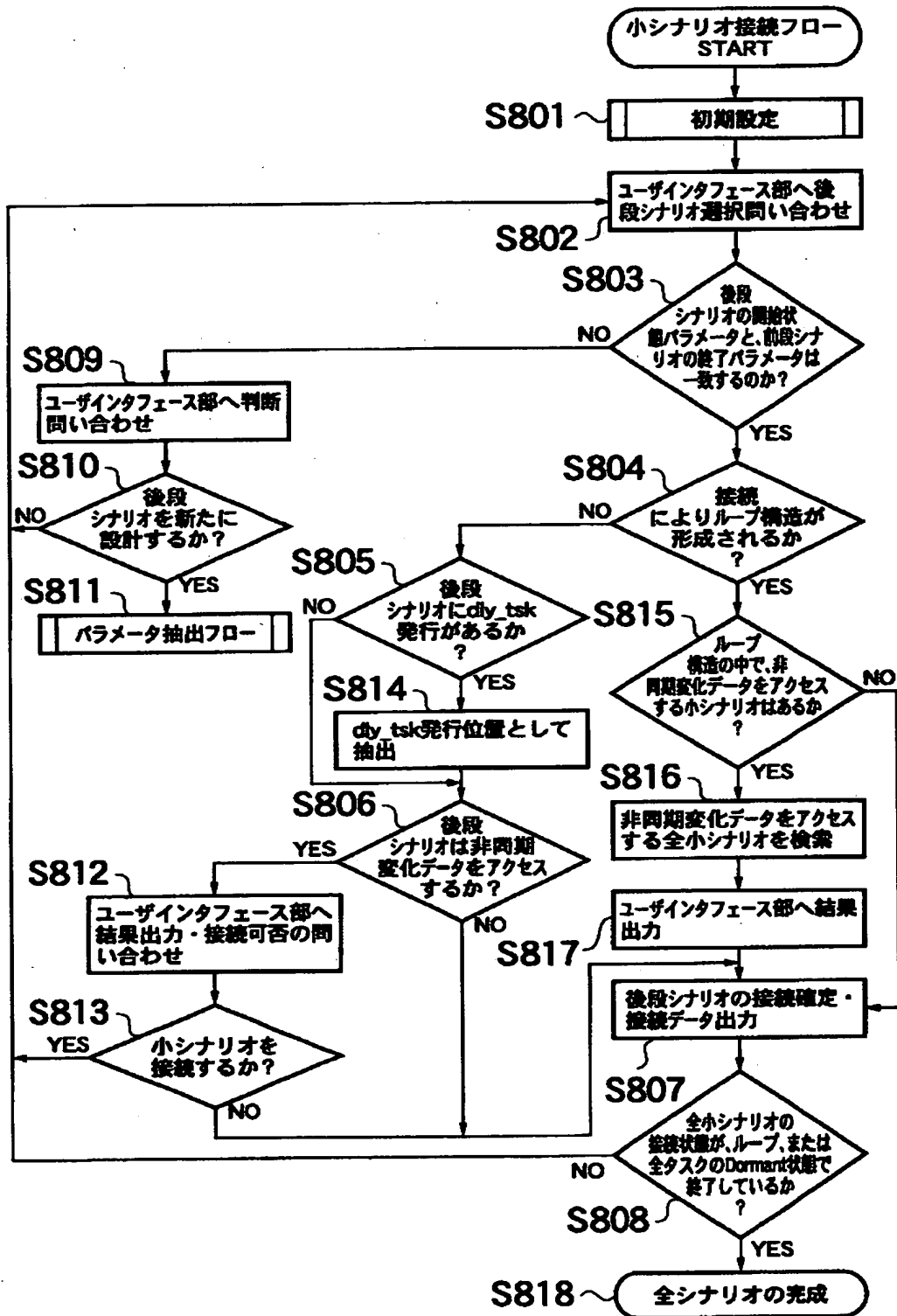
【図 9】



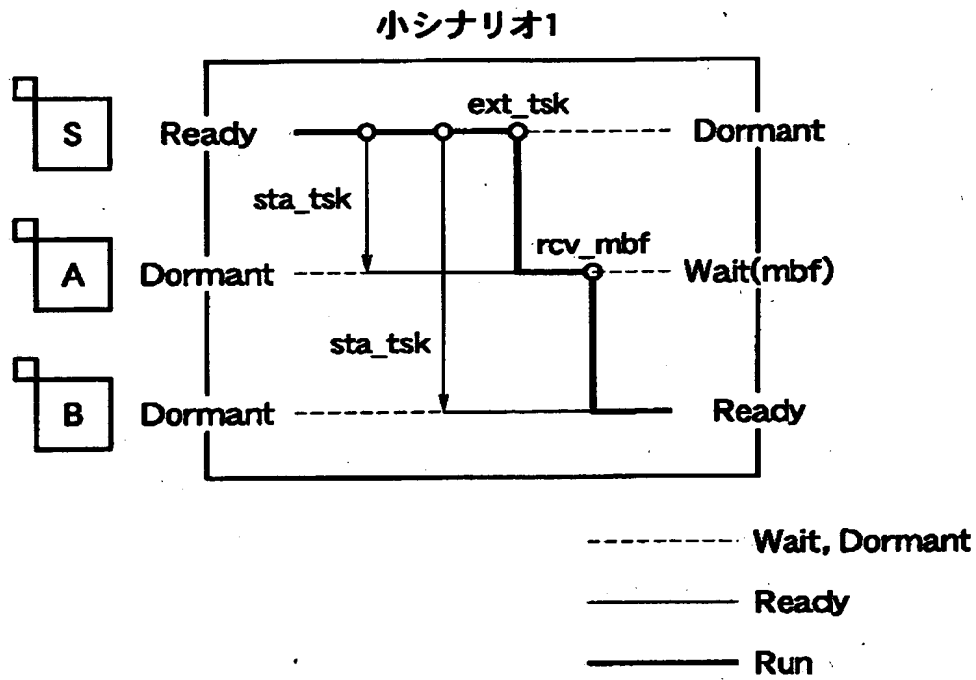
【図10】



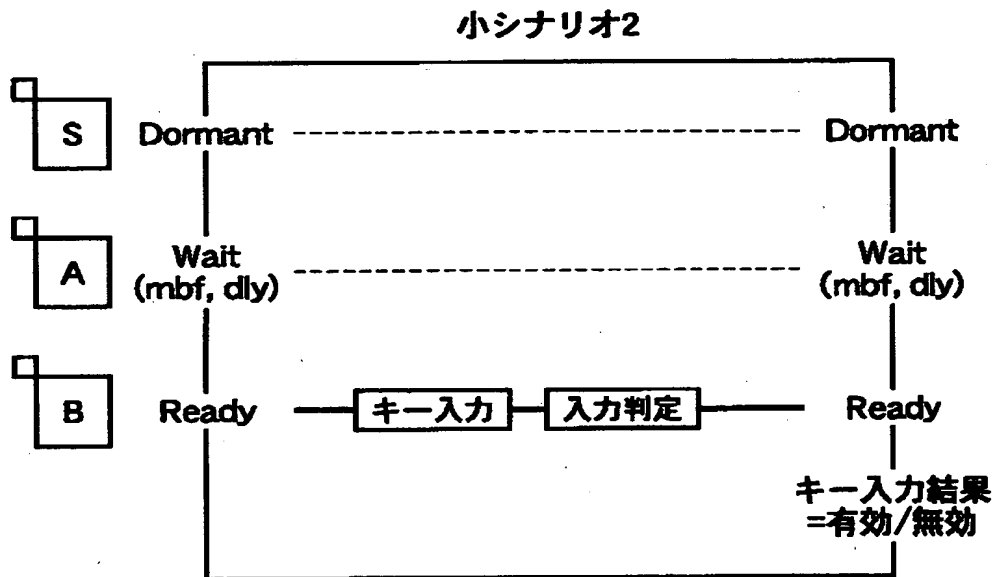
【図 11】



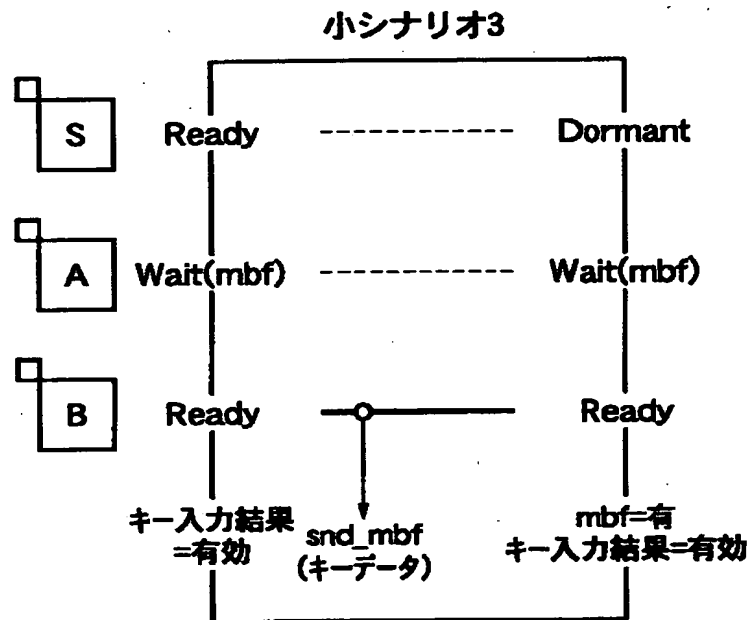
【図 1 2】



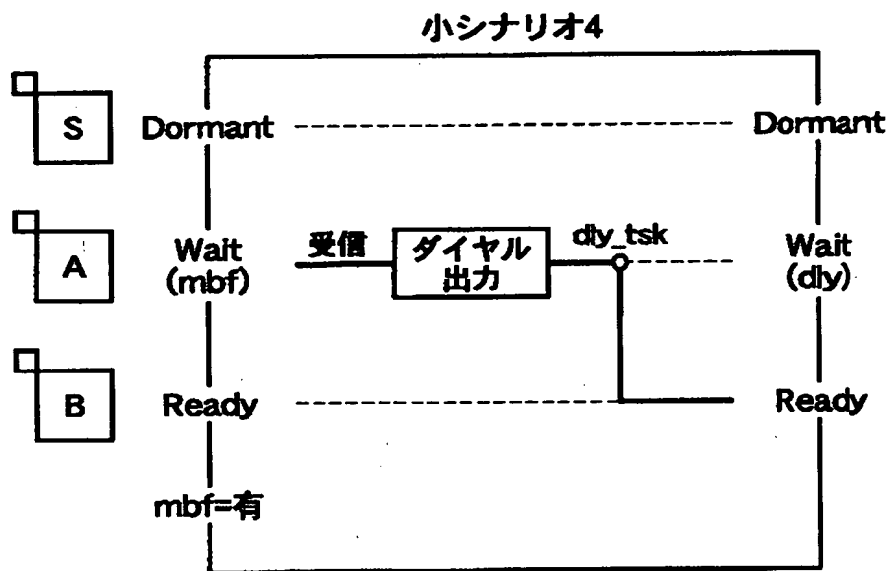
【図 1 3】



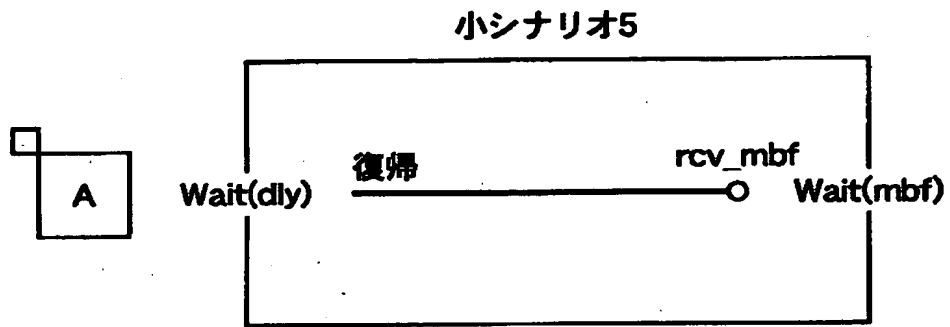
【図14】



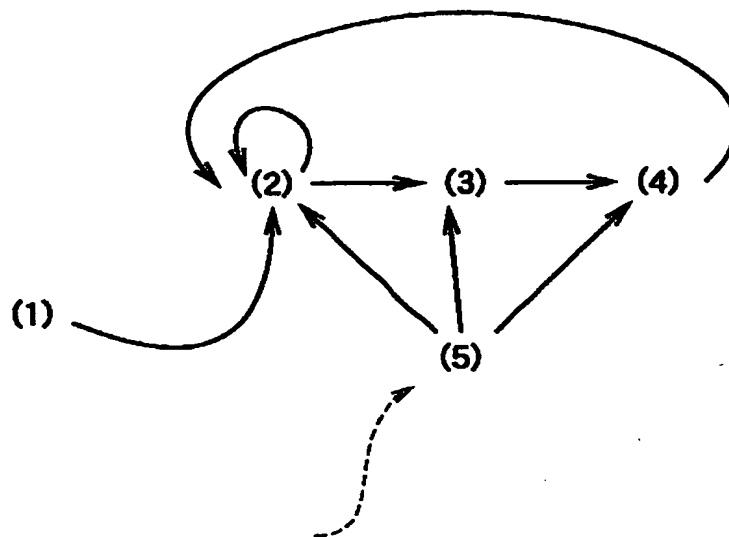
【図15】



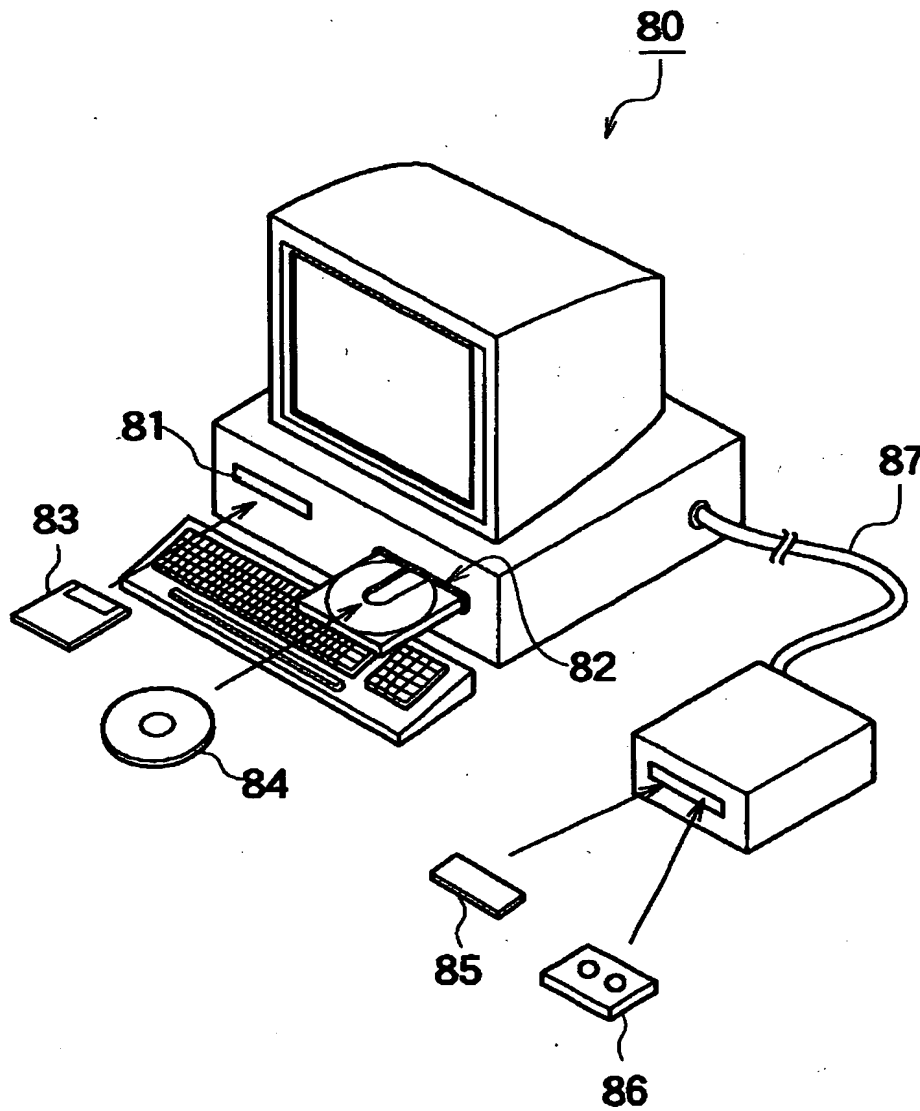
【図 1 6】



【図 1 7】



【図18】



【書類名】 要約書

【要約】

【課題】 リアルタイムシステムの構築を容易にし、リアルタイムシステムを使ったアプリケーションの開発期間を短縮する。

【解決手段】 ユーザの入力データを受け付け、またユーザにパラメータ抽出結果等を示すユーザインタフェース部11と、小シナリオの開始状態又は終了状態を示すパラメータ等を抽出する小シナリオパラメータ抽出部12と、抽出された各種パラメータを記憶する小シナリオ状態情報記憶部14と、2つの小シナリオを接続した際にシステムが矛盾した動作を起こさないかを判断する小シナリオ接続部・整合性判断部13と、小シナリオ間の接続データを記憶する小シナリオ間接続データ記憶部15とを具備する。

【選択図】 図4

出 願 人 履 歴 情 報

識別番号 [000003078]

1. 変更年月日 1990年 8月22日

[変更理由] 新規登録

住 所 神奈川県川崎市幸区堀川町72番地

氏 名 株式会社東芝

出 願 人 履 歴 情 報

識別番号 [598010562]

1. 変更年月日 1998年 1月23日

[変更理由] 新規登録

住 所 神奈川県川崎市幸区堀川町580番地

氏 名 東芝エルエスアイシステムサポート株式会社